CLAIMS

1. A controller for a random access memory comprises:

an address and command queue that holds memory references from a plurality of microcontrol functional units;

a first read/write queue that holds memory reference from a computer bus;

a second read/write queue that holds memory references from a core processor; and

control logic including an arbiter that detects the fullness of each of the queues and a status of outstanding memory reference to select a memory reference from one of the queues.

2. The controller of claim 1 wherein the control logic further selects one of the queues to provide a next memory references based on a programmable value stored in a priority service control register.

3. The controller of claim 1 wherein the address and command queue comprises:

a high priority queue that holds memory references from high priority tasks.

4. The controller of claim 1 wherein the address and command queue comprises:

an even bank queue;

an odd bank queue; and

wherein a microengine sorts memory references into odd bank and even bank references.

5. The controller of claim 1 wherein the address and command queue comprises:

an order queue and wherein controller examines an optimized memory reference bit and if set, causes incoming reference requests to be sorted into either the even bank queue or the odd bank queue.

6. The controller of claim 5 wherein the address and command queue comprises:

an order queue; and

wherein if the memory reference request does not have a memory optimization bit set, the memory reference is stored in the order queue.

7. The controller of claim 1 wherein the address and command queue is implemented in a single memory structure and comprises:

an order queue for storing memory references;

an even bank queue for storing memory references;

an odd bank queue for storing memory references;

a high priority queue for storing memory references; and

with the memory structure being segmented into four different queue regions, each region having its own head and tail pointer.

8. The controller of claim 7 wherein the address and command queue further comprises:

an insert queue control and a remove queue arbitration logic to control insert and removal of memory references from the queues.

9. The controller of claim 1 further comprising:

a command controller and address generator that is

responsive to an address from a selected memory reference from one said queues, to produce addresses and commands to control a memory interface.

10. The controller of claim 9 further comprising:

a memory interface responsive to generated addresses and commands to produce memory control signals.

11. The controller of claim 1 wherein the control logic is responsive to a chaining bit that when set allows for special handling of contiguous memory references.

12. The controller of claim 1 wherein assertion of the chain bit will control the arbiter to have the arbiter select the functional unit which previously requested that bus because setting of the chain bit indicates that the microengine issued a chain request.

13. The controller of claim 1 wherein the control logic is responsive to an optimized memory bit and a chaining bit, and wherein assertion of the chaining bit will control the arbiter when the optimized memory bit is also set to maintain the memory references from a current queue.

14. The controller of claim 1 wherein the arbiter has an arbitration policy that favors chained microengine memory references.

15. The controller of claim 14 wherein the arbiter has an arbitration policy that services chained requests until the chain bit is cleared.

16. The controller of claim 1 wherein the arbiter has an arbitration policy that starts by examining for chained microengine memory reference requests.

17. The controller of claim 16 wherein the arbitration policy enables chained memory requests to be serviced completely.

18. The controller of claim 16 wherein when the chain bit is set, the arbitration engine services the same queue again until the chain bit is cleared.